



(43) International Publication Date
3 October 2013 (03.10.2013)

(10) International Publication Number
WO 2013/149155 A2

- (51) International Patent Classification:
G06K 9/46 (2006.01)
- (21) International Application Number:
PCT/US2013/034624
- (22) International Filing Date:
29 March 2013 (29.03.2013)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
13/433,429 29 March 2012 (29.03.2012) US
- (71) Applicant: **ANALOG DEVICES, INC.** [US/US]; One Technology Way, Norwood, MA 02062-9106 (US).
- (72) Inventor; and
(71) Applicant : **MATH, Roopa, S.** [IN/IN]; No. 9, 1st Cross GH Layout, 14th Main, Jayanagar 3rd Block, Bangalore 560011 (IN).
- (74) Agent: **RUSSELL, Steven, J.**; Bingham Mccutchen LLP, 2020k Street, Nw, Washington, DC 20006 (US).
- (81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

- *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) Title: INCREMENTAL CONTOUR-EXTRACTION SCHEME FOR BINARY IMAGE SEGMENTS

(57) Abstract: Contours of objects appearing in a digital image are extracted in a plurality of one-directional passes across the digital image. Each pass loads rows or columns of the image into a local memory, in the order they appear in the image, and analyzes them for the presence of portions of the full contour. The portions are then combined to create the full contour.



WO 2013/149155 A2

INCREMENTAL CONTOUR-EXTRACTION SCHEME FOR BINARY IMAGE SEGMENTS

RELATED APPLICATIONS

5 This application claims the benefit of and priority to U.S. Patent Application No. 13/433,429 filed March 29, 2012, the entire disclosure of each of which is hereby incorporated herein by reference.

TECHNICAL FIELD

10 [0001] Embodiments of the invention generally relate to digital-image processing and, in greater detail, to detecting contours in images.

BACKGROUND

15 [0002] Finding a contour of an object (i.e., the object's border or outline) in a digital image is often the first step in many image-analysis applications, such as object detection or object classification. Given a preprocessed version of the image, many current contour-extraction methods identify an edge pixel and search the area of the image nearby for another, connected edge pixel. The area searched may include the four pixels connected to a side of the current edge pixel (i.e., the "four-connected neighbors") or the eight pixels connected via both sides and corners (i.e., the "eight-connected neighbors"). By repeatedly searching for more connected edge pixels, a contour may be traced by traversing the edge, wherever in the image it leads, in a clockwise or counter-clockwise direction. As more edge pixels are found, they may be added to a list of pixels belonging to the contour. This process is continued until the first identified edge pixel is encountered again (i.e., the entire contour is traced).

20 [0003] While this technique may find the contours present in an image, it may run quite slowly on some processors. Because the edge path may travel in any direction, and because this direction cannot be predicted, pixels from any portion of the image may need to be analyzed as the algorithm works. Many processors (especially low-power or digital-signal processors) are unable to store an entire image in local, fast memory and must therefore load necessary portions of the image from external, slow memory as they are needed. These memory accesses may significantly retard the operation of the algorithm. Other algorithms may process the digital image in a more deterministic fashion (e.g., by marching across the pixels in the image), but these methods are typically incapable of fully identifying the contours of complex shapes. A need therefore exists for a way to detect

30

- 5 contours of arbitrary shapes in a digital image in a manner that runs efficiently on a processor having a small local memory.

SUMMARY

[0004] In general, various aspects of the systems and methods described herein extract contours of shapes within digital images by making a maximum of deterministic and predictable passes —
10 e.g., four such passes — across the image. The passes may include or consist of a top-to-bottom pass, a bottom-to-top pass, a left-to-right pass, and/or a right-to-left pass. Because each pass proceeds row-by-row (or column-by-column) across the image, soon-to-be needed rows or columns may be prefetched from an external memory into a local, fast memory before they are needed, thus saving delays incurred by random accesses to the external memory. The multiple passes may each
15 extract a portion of the full contour of a shape; the passes proceed until the portions may be combined to create the full contour. The combination and/or checking of the portions may proceed in two directions simultaneously, thus providing a further speed benefit. In one embodiment, during each pass, an extracted left contour pixel is added to the left-most part of a list of discovered contour pixels and an extracted right contour pixel is added to the right-most part of the list of
20 contour pixels to ensure that, at any given time, the list of contour pixels is ordered in accordance with the location of the pixels in the contour (i.e., reading the list in order represents an only one-directional movement along the contour). The direction may be clockwise or counter-clockwise; this ordering makes it easier to join the pieces of the contour extracted over different passes.

[0005] In one aspect, a method for extracting contours in a digital image includes performing a
25 plurality of one-directional passes across the digital image. Each pass includes (i) loading rows or columns of the digital image from an external memory to a local memory, (ii) identifying pixels in each row or column that correspond to a new contour of an object in the digital image or to a continuation of an already-found contour of the object, and (iii) adding the identified pixels to a list of contour pixels corresponding to the object for the current pass. The list of contour pixels from
30 each pass is combined into a full contour of the object.

[0006] The digital image may be a binary digital image. The plurality of one-directional passes may include of two, three, or four one-directional passes; each pass may include a top-to-bottom pass, a bottom-to-top pass, a left-to-right pass, and/or a right-to-left pass. Combining the list of contour pixels from each path may include searching for an endpoint of a first contour portion in a
35 second, adjacent contour portion and connecting the first and second contour portions at the endpoint. Identifying pixels in each row or column that correspond to the new contour may include identifying an object at a position in a current row or column; the position may be unoccupied by

5 the object at a corresponding position in a previous row or column. Identifying the object at the position in the current row or column may include identifying a binary “1” digit and wherein the corresponding position in the previous row or column held a binary “0” digit. Identifying pixels that correspond to a continuation of an already-found contour may include (i) finding a beginning pixel or an end pixel of the already-found contour in a previous row or column and (ii) searching
10 for a neighboring contour pixel in a current row or column.

[0007] In another aspect, a system for extracting contours in a digital image includes a local memory for storing data corresponding to the image and a processor; the processor includes a memory management unit and an execution unit. The memory management unit loads a portion of the image from an external memory to the local memory. The execution unit executes instructions
15 that (i) perform a plurality of one-directional passes across the digital image by analyzing the portion of the image, each pass identifying a portion of a contour of an object in the digital image and (ii) combine the portions of the contour into a full contour of the object.

[0008] The processor may be a digital-signal processor, and a size of the local memory may be less than the size of the image. The portions of the image loaded from the external memory may
20 include rows or columns of the image. A high-speed interface may connect the local memory to the processor and a low-speed interface may connect the external memory to the processor. The instructions to combine the portions of the contour may include instructions to search for an endpoint of a first contour portion in a second, adjacent contour portion and to connect the first and second contour portions at the endpoint. The plurality of one-directional passes may include of
25 two, three, or four one-directional passes, and may include a top-to-bottom pass, a bottom-to-top pass, a left-to-right pass, and/or a right-to-left pass.

[0009] These and other objects, along with advantages and features of the present invention herein disclosed, will become more apparent through reference to the following description, the accompanying drawings, and the claims. Furthermore, it is to be understood that the features of the
30 various embodiments described herein are not mutually exclusive and can exist in various combinations and permutations.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] In the drawings, like reference characters generally refer to the same parts throughout the different views. In the following description, various embodiments of the present invention are
35 described with reference to the following drawings, in which:

[0011] FIG. 1 illustrates a system for extracting contours in accordance with an embodiment of the invention;

5 [0012] FIG. 2 illustrates a method for extracting contours in accordance with an embodiment of the invention;

[0013] FIG. 3A illustrates a shape appearing in a binary image;

[0014] FIG. 3B illustrates a partial contour of the shape extracted in a first pass in accordance with an embodiment of the invention;

10 [0015] FIG. 3C illustrates a partial contour of the shape extracted in a second pass in accordance with an embodiment of the invention; and

[0016] FIG. 3D illustrates a full contour of the shape combined from partial contours in accordance with an embodiment of the invention.

DETAILED DESCRIPTION

15 [0017] Described herein are various embodiments of methods and systems for extracting contours of shapes appearing in digital images. A system 100 configured in accordance with an embodiment of the invention is illustrated in FIG. 1. A processor 102 executes stored instructions in an execution unit 104 to extract the contour. In one embodiment, a local memory 106 is located near the processor 102 and is connected to it via a high-speed interface 108; data in the local memory
20 106 may be accessed quickly and efficiently by the processor 102. The local memory 106 may be located physically adjacent to the processor 102 (as shown in FIG. 1) and configured as, for example, a local SRAM or cache memory. In another embodiment, the local memory 106 is located within the processor 102 as an on-chip cache (e.g., L1 or L2 memory), register file, or any other type of high-speed storage. The processor 102 may be any type of digital processor, such as a
25 microprocessor, low-power processor, digital-signal processor, or ASIC. The current invention is not, however, limited to any particular type of processor 102 or local memory 106.

[0018] In one embodiment, the processor 102 extracts one or more contours of objects in a digital image having a size larger than the maximum storage capacity of the local memory 106. In this embodiment, the entire image (or the remainder of the image not otherwise stored in the local
30 memory 106) is stored in an external memory 110, which is connected to the processor 102 via a low-speed interface 112 (i.e., the speed/bandwidth of the interface 112 to the external memory 110 is less than the speed/bandwidth of the interface 108 to the local memory 106). As explained in greater detail below, the processor 102 (using, e.g., a memory-management unit 114) may pre-fetch portions of the image from the external memory 110 before they are needed, thereby avoiding any
35 delay incurred by use of the low-speed interface 112. Embodiments of the invention also operate on systems having large local memories (i.e., memories large enough to store an entire digital image); the current invention is not limited to only systems having small local memories.

5 **[0019]** A method 200 for extracting contours of shapes appearing in binary images is illustrated in FIG. 2. In brief overview, a series of one-directional passes is performed across the image and, after the completion of each pass, a test is performed to see if a full contour has been traced around any found objects. Up to four passes may be performed: top-to-bottom, bottom-to-top, left-to-right, and/or right-to left, and the performed passes may occur in any order. Each pass includes a one-
10 directional march across the image; the march is by rows of the image in the case of top-to-bottom or bottom-to-top passes and by columns of the image in the case of left-to-right or right-to-left passes. Once a row or column is considered and analyzed, it may be discarded from the local memory 106 to thereby make room for additional rows or columns. In one embodiment, one row or column is analyzed at a time; while a first row or column is being analyzed, the next row or
15 column may be pre-fetched from the external memory 110. More than one row or column may be analyzed at once, however, and multiple rows or columns may be pre-fetched at a time.

[0020] In greater detail, each pass 202 through the digital image includes pre-fetching rows or columns (step 204), identifying pixels in each row or column that belong to a contour of an object in the image (step 206), and storing any identified pixels (step 208) in, e.g., a list or data structure.

20 In the pre-fetching step 204, the rows or columns may be fetched in order, beginning with the row or column at a first edge of the image and proceeding across the image to the far edge. The pre-fetching may be timed such that the rows or columns arrive at the processor 102 as they are needed for processing by the later steps in the pass. Thus, the pre-fetching may be timed in accordance with the latency of the external memory 110 and its interface 112. If the local memory 106 is large
25 enough, several rows or columns may be pre-fetched in a single operation and stored therein, thus allowing the interface 112 to be free for other memory transactions. If the local memory 106 is too small to store a single row or column, only a portion thereof may be pre-fetched at once. In one embodiment, a left-to-right or right-to-left pass is preferred over a top-to-bottom or bottom-to-top pass (or vice versa) if the digital image is rectangular and one dimension (e.g., a column) fits inside
30 the local memory 106 while the other dimension (e.g., a row) is too large to fit.

[0021] Once a row or column (or portion thereof) is available to the processor 102, it is analyzed for pixels corresponding to a contour of an object. In one embodiment, the digital image is a binary image, in which each pixel is represented by a binary “1” or “0.” This binary image may be produced by, for example, analyzing the pixel values of the original digital image by intensity or
35 color value and assigning a “1” if a pixel’s intensity or color is greater than a threshold and a “0” otherwise. Any method of creation of the binary image is within the scope of the current invention, however. An example of a binary image is shown in FIG. 3A and discussed in greater detail below. In other embodiments, the processor analyzes raw image data and assigns a “1” or “0” value (or their equivalents) on the fly.

5 **[0022]** The first row or column of the binary image is searched for “1” pixels. In one embodiment, any such values found in the first row or column are designated as part of a contour of an object by virtue of their being on the edge of the digital image; in other embodiments, objects coincident with an edge of the digital image are not deemed to have a contour on that edge. The found “1” pixels that correspond to a contour are (like other such pixels found in later rows or
10 columns) saved in a data structure (e.g., a list). For example, the (x,y) position (i.e., row and column number) of each contour pixel may be saved. If a series of adjacent “1” pixels is found (referred to herein as a “run” of pixels), the beginning and end points of the run of pixels may be determined and stored.

15 **[0023]** In one embodiment, a contour-node data structure stores information about the contour of an object. The contour-node data structure may include a plurality of (x,y) value pairs that represent either pixels in the contour or runs of pixels in the contour (referred to, collectively, as “nodes” of the contour). In another embodiment, instead of (x,y) value pairs, information about the contour nodes is stored in other formats (such as a Freeman code). Each node in the data structure may be associated with a pointer to a next node so that a contour may be traced from node to node.
20 An “in-use” flag may also be associated with each node to indicate that the node is part of a current contour (and thus should not be changed).

25 **[0024]** Another data structure may be used to store information corresponding to a run of pixels (as referred to above). This run-length data structure may include, for each run of pixels, x-coordinates for the beginning and end points of the run of pixels and the y-coordinate of the run of pixels. These coordinates may be swapped appropriately for left-to-right or right-to-left passes through the digital image. Further information may be associated with each run of pixels, such as (i) if one, both, or neither of the ends of the run are currently connected to other nodes in the contour, (ii) a pointer to the next node in the contour, and/or (iii) if portions of the contour are extracted in multiple passes across the image, the portion to which the run belongs and a pointer to
30 the next portion.

35 **[0025]** Information related to the portion of the contour extracted in a single pass may be stored in yet another data structure. This contour-header data structure may include the direction of the pass, the first contour node found for the portion, and the rightmost and leftmost nodes in the portion. Other information may include a pointer to additional pieces of the portion of the contour and/or a flag indicating that the portion is complete and should not be further changed.

40 **[0026]** Returning to the pass through the image, as subsequent rows or columns are pre-fetched and analyzed, additional “1” pixels may be found. Unlike the first row or column, however, not all of these “1” pixels may correspond to a contour, because some of them may correspond to interior portions of an object. To distinguish between these two types of “1” pixels, data from the previous

5 row or column is analyzed and, in particular, the location of “1” pixels in the previous row or column may be retrieved from the data structure. If a “1” pixel is found at a certain position in a current row or column, but there was a “1” pixel at the corresponding position in the previous row or column, the current “1” pixel may be deemed to be an interior point in a found object and not stored in the data structure. In another embodiment, current “1” pixels may be tested to determine
10 if they lie between the beginning and endpoints of any runs of pixels in the previous row or column. If not, and/or if there is no “1” pixel in a corresponding position in the previous row or column, the current “1” pixel may be deemed to be the leading edge of a newly found object, and thus saved as a new contour.

[0027] The positions of the beginning and end points of any runs found in the previous row or
15 column may be examined in the current row or column to determine the new positions of the beginning and endpoints corresponding to each object. These positions may be found by searching the neighborhood of the previous beginning point for a zero-to-one transition in the current row or column and by searching the neighborhood of the previous end point for a one-to-zero transition in the current row or column. For example, if the object under consideration is a square, the positions
20 of the beginning and end points remain the same in the current row or column as in the previous row or column, and thus the zero-to-one and one-to-zero transitions occur directly underneath the previous beginning and end points. If, however, the object under consideration changes its shape in the new row or column (as would, e.g., a triangle, trapezoid, or other similar shape), the zero-to-one and one-to-zero transitions may occur at new positions. In this case, the neighborhood of the
25 previous beginning and end points is searched in the current row or column until the transitions, and thus the new points, are found. The new beginning and end points are saved in the data structure as part of its contour.

[0028] In one embodiment, each new row or column is analyzed in a single pass from a beginning pixel to an ending pixel, thereby speeding execution time. For example, each “1” pixel, runs of “1”
30 pixels, and zero-to-one and one-to-zero transitions are detected in a single pass, and a determination is made during that pass which “1” pixels correspond to contours and which correspond to object interiors. The current invention is not limited to only this implementation, however, and any method of analyzing each row or column is within the scope of the current invention.

[0029] When the last row or column in the image is analyzed, the pass through the image is
35 complete. The stored contour pixels may then be analyzed to determine if they represent one or more full contours (i.e., if they represent closed paths). In one embodiment, the entire contour of an object may be determined in a single pass, and the method exits. In other embodiments, however, the objects are too complicated to have their contours extracted in a single pass. For example, while the above method may extract a portion of a contour of an object having a concave

5 edge facing the direction of the pass, it may not extract a portion of a contour of an object having a concave edge facing away from the direction of the pass at least because this away-facing concave edge, at the time of its discovery, may be part of the exterior contour or an interior contour. The type (interior/exterior) of the contour may be determined only when the pass reaches the end of image (i.e., the last row or column). For example, given a top-to-bottom pass, if the away-facing
10 concave edge is like that of the bottom part of an “H” shape, then it is indeed part of the external contour of the shape. On the other hand, however, if the shape resembles an inverted “A”, the detected contour corresponding to the crossbar of the “A” is part of an interior contour; which is realized only when the end/tip of the “A” is reached. A subsequent pass (e.g., a bottom-to-top pass) may be needed to decide its inclusion into the external contour. Thus, a second, third, or
15 fourth pass may be necessary to complete the full contour of the object.

[0030] With reference again to FIG. 2, when additional passes are complete, any contour pixels found in the additional passes are merged with the contour pixels found in previous passes (step 210). The completeness of the merged contour pixels is analyzed (step 212) and, if a complete contour is not found, another pass is performed (step 214). If a full contour is found after two,
20 three, or four passes, the process ends (step 216).

[0031] To check for the completeness of a contour, existing systems may chose a point on the contour trace it in a clockwise or counter-clockwise direction (depending on the way the contour pixels are stored). If the process finds its way back to the original point after traversing the contour, the contour is deemed complete. This traversing may require the re-visiting of previously
25 examined rows or columns in the contour, however. For example, if the object is an “H” shape and the traverse is clockwise, the rows in the contour are loaded and analyzed once on the way down the right-hand side of the “H” shape and again on the way up on the left-hand side of the “H” shape. In one embodiment, the contour pieces are maintained in such a way that there is no need to trace both sides – it is enough to search in just one direction. For example, each contour piece may
30 be maintained such that its rightmost and leftmost pixels are accessible (e.g., there are fields in a data structure that point to those pixels). A first contour piece may be have leftmost and rightmost pixels C1:Left and C1:Right and a second contour piece may have leftmost and rightmost pixels C2:Left and C2:Right. The rightmost pixel C1:Right in the first contour piece is accessed; the second contour piece is then searched in a clockwise direction, from its leftmost pixel C2:Left, until
35 a pixel having the same coordinates as C1:Right is found. The searched pixels in the second contour piece (i.e., the pixels between C2:Left and C1:Right) thus correspond to an “overlapping” portion of the contour that appears in both the first and second contour pieces; this overlapping portion is discarded. Similarly to the above operation, the rightmost pixel C2:Right of the second contour is accessed and the first contour is searched from its leftmost pixel C1:Left (again in a

clockwise direction); once the corresponding pixel is found, the overlapping region between C1:Left and C2:Right is also discarded. The non-discarded sections of the first and second contour portions are then combined to create a full contour of the object. In other embodiments, the search may be done in a counter-clockwise direction and/or more than two contour pieces may be stitched together.

[0032] An example binary image 300 is shown in FIG. 3A. The image is 14×14 pixels and includes a shape 302 that resembles the letter “H.” Notably, the shape 302 has an upward-facing concave side 304 and a downward-facing concave side 306. FIG. 3B illustrates a result 308 of a top-to-bottom pass through the image 300 in accordance with an embodiment of the invention. As described above, the “1” pixels 310 in row 4 of the image 300 are deemed to be part of a contour because there were no corresponding “1” pixels in row 3 of the image 300; thus, as the “1” pixels in row 4 were encountered in the top-to-bottom pass, they were added to the contour of the object 302. As further rows 5–7 of the image 300 are analyzed, further contour pixels 312, corresponding to vertical portions of the original shape 302, are discovered in accordance with the current invention. As described above, the neighborhoods of the beginning and end points of the runs of pixels 310 are searched in later-analyzed rows to find further contour pixels (i.e., the pixels 312).

[0033] Still referring to FIG. 3B, when the pass through the object 302 reaches the first concave edge 304 (in row 8 of the original image 300), it discovers its corresponding contour portion 314. This new portion of the contour 314 connects the previously discovered portions 310, 312 of the contour, which, prior to reaching the first concave edge, appeared as if they were two separate objects. The portion 314 of the contour corresponding to the concave edge 304 may be detected by analyzing the absence of “1” pixels in corresponding positions in the previous row 7, by searching for zero-to-one or one-to-zero transitions (or the lack thereof) in the neighborhood of the beginning and end points of the runs of pixels 312, or by a combination of the two techniques. The remainder of the object 302 is scanned; the downward-facing concave edge 306 is not detected in this pass.

At the end of the top-to-bottom pass, the extracted contour 308 is analyzed for completeness, and because it does not create a full path, another pass is performed.

[0034] FIG. 3C illustrates a result 316 of a second, bottom-to-top pass of the image 300. The result 316 includes another part of the contour of the object 302; in this pass, the bottom-facing concave edge 306 is detected while the top-facing concave edge 304 is not (for reasons similar to the top-to-bottom pass described above). At this point, however, the two partial contours 308, 316 may be combined; the result 318 of their combination is shown in FIG. 3C. The resultant contour 318 is tested for completeness and, because it includes a full path around the exterior of the original object 302, no further passes are required, and the process halts.

5 **[0035]** It should also be noted that embodiments of the present invention may be provided as one or more computer-readable programs embodied on or in one or more articles of manufacture. The article of manufacture may be any suitable hardware apparatus, such as, for example, a floppy disk, a hard disk, a CD ROM, a CD-RW, a CD-R, a DVD ROM, a DVD-RW, a DVD-R, a flash memory card, a PROM, a RAM, a ROM, or a magnetic tape. In general, the computer-readable programs
10 may be implemented in any programming language. Some examples of languages that may be used include C, C++, or JAVA. The software programs may be further translated into machine language or virtual machine instructions and stored in a program file in that form. The program file may then be stored on or in one or more of the articles of manufacture.

[0036] Certain embodiments of the present invention were described above. It is, however,
15 expressly noted that the present invention is not limited to those embodiments, but rather the intention is that additions and modifications to what was expressly described herein are also included within the scope of the invention. Moreover, it is to be understood that the features of the various embodiments described herein were not mutually exclusive and can exist in various combinations and permutations, even if such combinations or permutations were not made express
20 herein, without departing from the spirit and scope of the invention. In fact, variations, modifications, and other implementations of what was described herein will occur to those of ordinary skill in the art without departing from the spirit and the scope of the invention. As such, the invention is not to be defined only by the preceding illustrative description.

What is claimed:

CLAIMS

- 1 1. A method for extracting contours in a digital image, the method comprising:
2 performing a plurality of one-directional passes across the digital image, each pass
3 comprising:
4 i. loading rows or columns of the digital image from an external memory to a
5 local memory,
6 ii. identifying pixels in each row or column that correspond to a new contour of
7 an object in the digital image or to a continuation of an already-found
8 contour of the object, and
9 iii. adding the identified pixels to a list of contour pixels corresponding to the
10 object for the current pass; and
11 combining the list of contour pixels from each pass into a full contour of the object.
- 1 2. The method of claim 1, wherein the plurality of one-directional passes consists of two,
2 three, or four one-directional passes.
- 1 3. The method of claim 1, wherein the plurality of one-directional passes comprises a top-to-
2 bottom pass, a bottom-to-top pass, a left-to-right pass, and a right-to-left pass.
- 1 4. The method of claim 1, wherein combining the list of contour pixels from each path
2 comprises searching for an endpoint of a first contour portion in a second, adjacent contour portion
3 and connecting the first and second contour portions at the endpoint.
- 1 5. The method of claim 1, wherein the digital image is a binary digital image.
- 1 6. The method of claim 1, wherein identifying pixels in each row or column that correspond to
2 the new contour comprises identifying an object at a position in a current row or column, wherein
3 the position was unoccupied by the object at a corresponding position in a previous row or column.
- 1 7. The method of claim 6, wherein identifying the object at the position in the current row or
2 column comprises identifying a binary "1" digit and wherein the corresponding position in the
3 previous row or column held a binary "0" digit.
- 1 8. The method of claim 1, wherein identifying pixels that correspond to a continuation of an
2 already-found contour comprises (i) finding a beginning pixel or an end pixel of the already-found

3 contour in a previous row or column and (ii) searching for a neighboring contour pixel in a current
4 row or column.

1 9. A system for extracting contours in a digital image, the system comprising:

2 a local memory for storing data corresponding to the image; and

3 a processor comprising:

4 i. a memory management unit for loading a portion of the image from an
5 external memory to the local memory, and

6 ii. an execution unit for executing instructions to (i) perform a plurality of one-
7 directional passes across the digital image by analyzing the portion of the
8 image, each pass identifying a portion of a contour of an object in the digital
9 image and (ii) combine the portions of the contour into a full contour of the
10 object.

1 10. The system of claim 9, wherein the portions of the image loaded from the external memory
2 comprise rows or columns of the image.

1 11. The system of claim 9, wherein the processor is a digital-signal processor.

1 12. The system of claim 9, further comprising a high-speed interface that connects the local
2 memory to the processor and a low-speed interface that connects the external memory to the
3 processor.

1 13. The system of claim 9, wherein a size of the local memory is less than the size of the image.

1 14. The system of claim 9, wherein the instructions to combine the portions of the contour
2 comprise instructions to search for an endpoint of a first contour portion in a second, adjacent
3 contour portion and to connect the first and second contour portions at the endpoint.

1 15. The system of claim 9, wherein the plurality of one-directional passes consists of two, three,
2 or four one-directional passes.

1 16. The system of claim 9, wherein the plurality of one-directional passes comprises a top-to-
2 bottom pass, a bottom-to-top pass, a left-to-right pass, and a right-to-left pass.

1 / 4

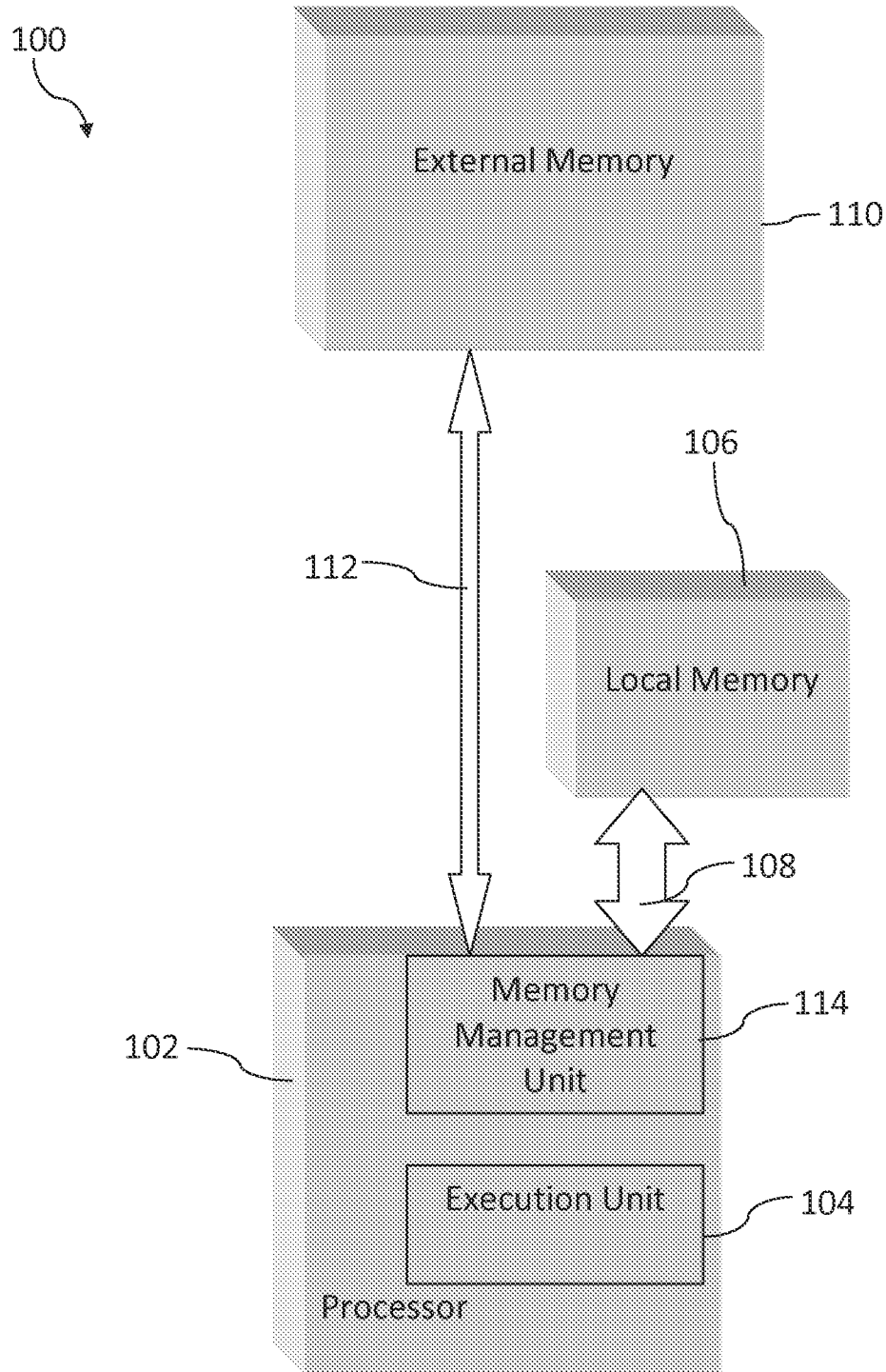


FIG. 1

2 / 4

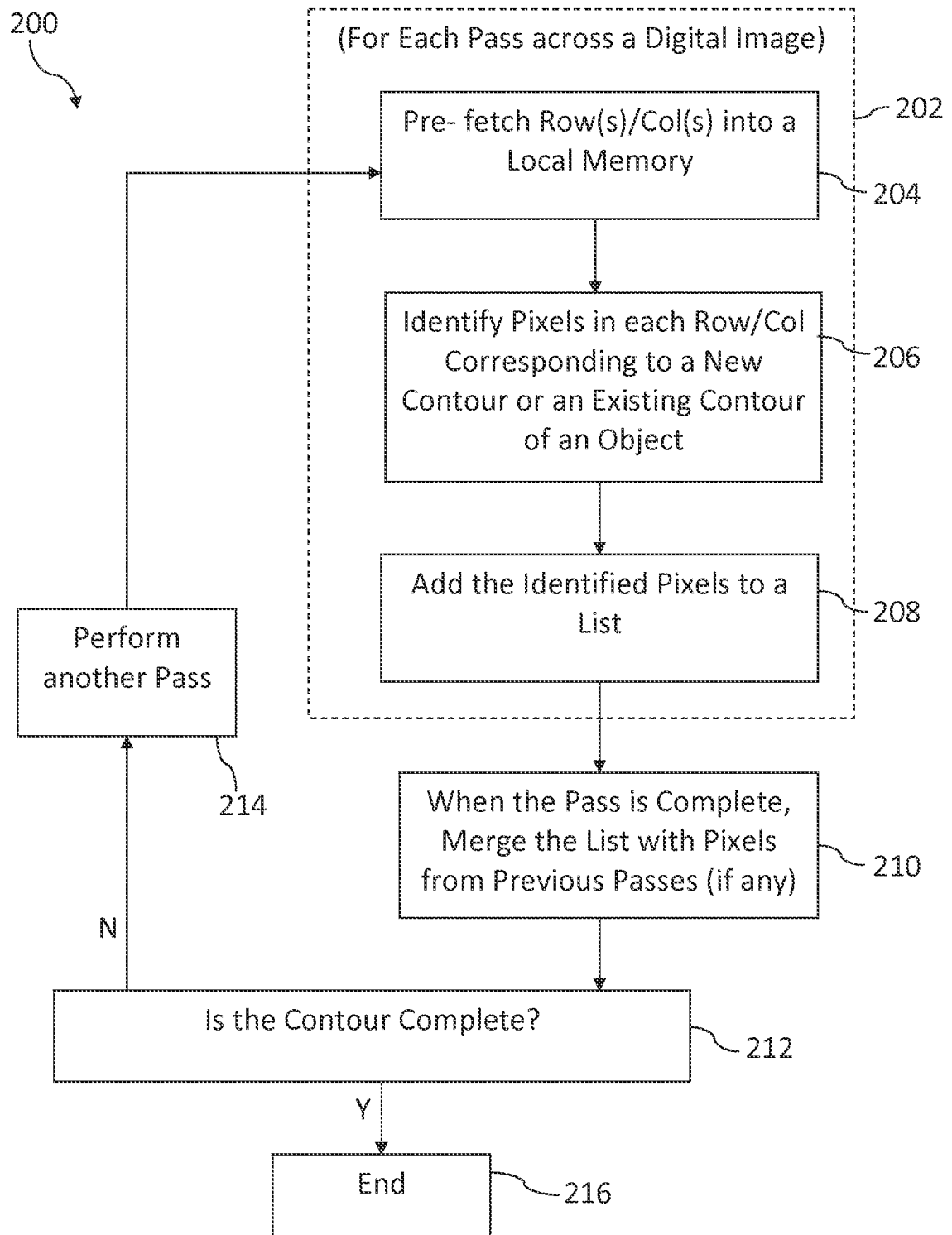


FIG. 2

[illegible]

FIG. 3A

[illegible]

FIG. 3B

4 / 4

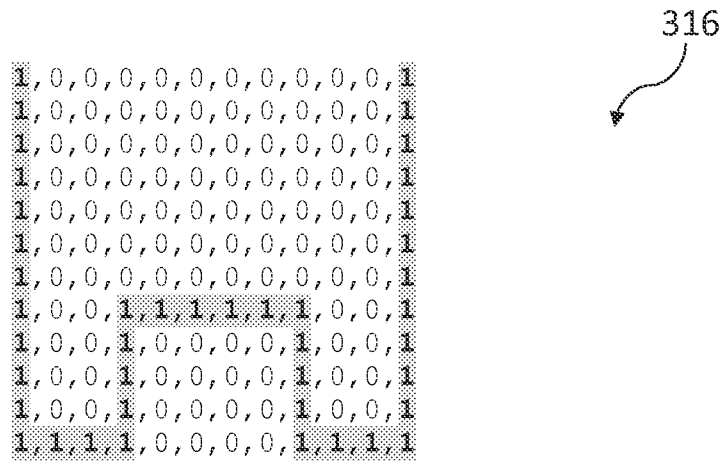


FIG. 3C

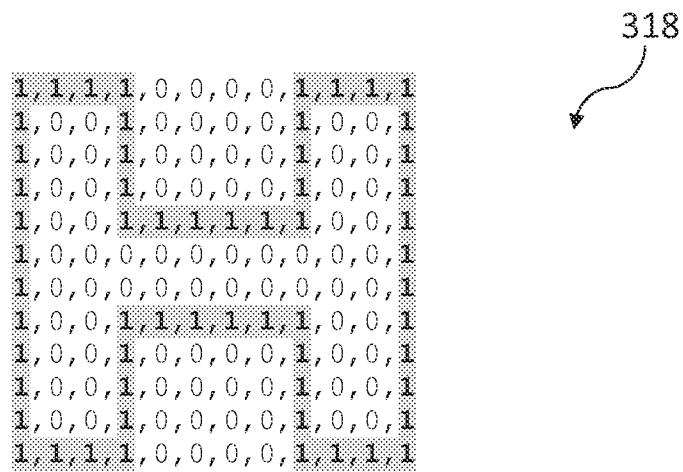


FIG. 3D